Amendments to the Specification:

- Please replace the original title with the following title:

MANAGEMENT OF SERVERS BY POLLING PROTOCOL DATA UNITS WITH MINIMIZED MANAGEMENT TRAFFIC AT DATA LINK LAYER

- Please amend the paragraph beginning on page 1, line 5 as follows:

The subject matter disclosed in each of the following applications is related: Rawson, ~~{Combining Network Management Information with Application Information on a Computer Network, Docket No. AUS920000520US;}~~ ***Network Management Server Combining PDUs to Minimize Bandwidth Consumption at Data Link Layer*****; Serial Number 09/799,361;** Rawson, ~~{Polling for and Transfer of Protocol Data Units in a Data Processing Network, Docket No. AUS920000516US1;}~~ ***Management of Servers by Polling Protocol Data Units with Minimized Management Traffic at Data Link Layer*****, Serial Number 09/799,358;** and Rawson, ~~{*Protocol Data Unit Prioritization in a Data Processing Network*, Docket No. AUS920000522US1}~~ ***Prioritization Of Network Management Server PDUs Versus Other PDUs at Data Link Layer*****, Serial Number 09/799,362**.

- Please delete the paragraph beginning on page 2, line 30.

~~In one embodiment, the second server NIC receives management PDUs from the first server and application PDUs from an external network. The NIC may be configured to interpret priority information in the management and application PDUs and enabled to prioritize interrupts to a host processor of the second server based upon the priority information. The management PDUs may be generated at a low level of the network's communication protocol stack. The communication protocol stack may comprise a TCP/IP protocol stack. The application PDUs are typically TCP/IP compliant while the management PDUs are generated at a data link level of the stack. The priority information may be contained within an IEEE 802.1q compliant header of the PDUs. The second server is typically configured to grant higher priority to application PDUs than management PDUs. The NIC may be configured to buffer management PDUs until a management PDU interrupt is issued. The second server NIC may be further configured to issue management PDU interrupts after detecting an absence of management PDU activity for a predetermined interval.~~

- Please amend the paragraph beginning on page 5, line 21 as follows:

Network 100 may further include a management server 110 connected to central switch 130. As its ~~{name, implies, management}~~ **name implies, management** server 110 is a dedicated server responsible for managing network 100 (including server appliances 101 and NAS 140). For purposes of this disclosure, typical management tasks include tasks associated with the deployment and configuration of server appliances 101, the installation of software and hardware upgrades, monitoring and managing network performance, security, failures, and storage capacity, and other statistical gathering and analysis tasks.

- Please amend the paragraph beginning on page 8, line 26 as follows:

Management PDUs typically comprise low-level data units transmitted between management server 110 and one or more device(s) that are locally connected on network 100. Management PDUs may travel between NIC 310 and the core 302 of management server 110 over different logical paths. Whereas application PDUs must typically flow through an entire protocol stack, management PDUs are typically generated at the lowest levels of a stack. Using the example of web based services running on a TCP/IP network, application PDUs are formatted to be TCP/IP compliant whereas management PDUs may have just a single header, such as a MAC header. FIG 4A illustrates an example of such a management PDU 401. As illustrated, management PDU 401 includes a management payload 412 and a MAC header 402 specifying the physical address of the ~~{PDUs}~~ PDU's target. Because management PDUs are destined for local targets, they do not require the Internet address and other information provided by the higher levels of the protocol stack.

- Please amend the paragraph beginning on page 9, line 7 as follows:

Processor 312 of NIC 310 is configured to detect PDUs generated by management server 110. Upon detecting a PDU, processor 312 may initially determine whether the PDU is a management PDU or an application PDU. An illustrative application PDU 403 is depicted in FIG 4B. In this example, application PDU 403 includes an application payload 410 and a TCP/IP compatible header structure including a TCP header 406, an IP header 404, and a MAC header 402. If the PDU is a management PDU, processor 312 may then determine whether there is an entry available in buffer 320 and store the management PDU in buffer 320 if there is an available entry. If there is no available entry in buffer 320, processor 312 may simply forward the management PDU to switch 130 without modification. Each management PDU typically includes payload information and a MAC header including a MAC address as discussed above indicating the server appliance or other device on network 100 to which the PDU is destined. Buffer 320 may be organized in a cache fashion where a ~~{PDUs}~~ PDU's MAC address is used as an index into buffer 320. In this embodiment, management PDUs are assigned entries in buffer 320 according to their MAC address. This arrangement may simplify the task of later determining if buffer 320 contains a PDU destined for a specific device. Alternatively, buffer 320 may comprise a random access memory in which a PDU associated with any target may be stored in any of the buffer entries.